

AMENDMENTS TO THE CLAIMS

1. (Currently amended) An information processing system, comprising:

a first computing device configured to:

~~receive an initialization packet originating from a client;~~

store an unbound data structure associated with a connection to ~~[[the]]~~ a client in response to receiving an initialization packet from the client;

select a computing device to service the client;

when the first computing device is selected to service the client, bind the unbound data structure ~~associated with a connection to the client~~ to an application of the first computing device; and

when ~~the first~~ a second computing device distinct from the first computing device is ~~[[not]]~~ selected to service the client, migrate the unbound data structure ~~associated with the connection~~ to the selected second computing device in a manner transparent to the client by outputting an encapsulated packet including the unbound data structure to the second computing device to enable the second computing device to output packets to the client configured to appear to the client to have been sent from the first computing device.

2. (Currently amended) The system of claim 1 wherein the ~~unbound data structure encapsulated packet~~ includes a group of sequence numbers associated with the connection.

3. (Currently amended) The system of claim 1 wherein the ~~bound data structure encapsulated packet~~ includes an IP address of the client, a port of an application executed by the client, an IP address of the first computing device, and a port of the application executed by the first computing device.

4. (Currently amended) An information processing system, comprising:
a first computing device configured to:

receive a request packet originating from a client;

when the packet is associated with a connection that corresponds to an application of the first computing device, forward the packet and a reference to an associated connection endpoint to a network protocol stack of the first computing device that is external to an operating system of the first computing device; and

when the packet is ~~[[not]]~~ associated with a connection that corresponds to an application of ~~the first~~ a second computing device distinct from the first computing device, selectively encapsulate the packet and forward the encapsulated packet to ~~[[a]]~~ the second computing device ~~in a manner transparent to the client~~, wherein the encapsulated packet includes a reference to a connection endpoint associated with the packet and an indication that the encapsulated packet is a forwarded packet.

5. (Previously presented) The system of claim 4 wherein the reference indicates a group of sequence numbers associated with the connection.

6. (Previously presented) The system of claim 4 wherein the reference indicates an IP address of the client, a port of an application executed by the client, an IP address of the first computing device, and a port of the application executed by the first computing device.

7. (Currently amended) The information processing system of claim 1 wherein in response to at least the initialization packet the first computing device is configured to generate an acknowledgement to the client before migrating the unbound data structure.

8-9. (Canceled)

10. (Currently amended) An information processing system, comprising:
a first computing device configured to:

associate an application of the first computing device with a data structure associated with a connection to a client; and, ~~in a manner transparent to the client,~~

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS[®]
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

selectively:

disassociate the application of the first computing device from the data structure; and

subsequently output an encapsulated packet including a reference to the data structure associated with the connection to a second computing device to associate an application of the second computing device with the data structure, and to enable the second computing device to output packets to the client configured to appear to the client to have been sent from the first computing device.

11. (Previously presented) The system of claim 10 wherein the data structure includes a group of sequence numbers associated with the connection.

12. (Previously presented) The system of claim 10 wherein the data structure includes an IP address of the client, a port of an application executed by the client, an IP address of the first computing device, and a port of the application executed by the first computing device.

13-15. (Canceled)

16. (Currently amended) A method performed by a first server, the method comprising:

storing an unbound data structure associated with a connection to a client originating an initialization packet;

selecting a server to service the client;

when the first server is selected to service the client, binding the data structure ~~associated with the connection to the client~~ to an application of the first server; and

~~when the first server is not~~ a second server distinct from the first server is selected to service the client, migrating the unbound data structure ~~associated with the connection~~ to the

selected second server in a manner transparent to the client by outputting an encapsulated packet including the unbound data structure to the second server to enable the second server to output packets to the client configured to appear to the client to have been sent from the first server.

17. (Currently amended) The method of claim 16 wherein the ~~unbound data structure encapsulated packet~~ includes a group of sequence numbers associated with the connection.

18. (Currently amended) The method of claim 16 wherein the ~~bound data structure encapsulated packet~~ includes an IP address of the client, a port of an application executed by the client, an IP address of the first server, and a port of an application executed by the first server.

19. (Currently amended) A method performed by a first computing device of an information processing system, the method comprising:

receiving a request packet originating from a client;

when the packet is associated with a connection that corresponds to an application of the first computing device, forwarding the packet and a reference to an associated connection endpoint to a network protocol stack of the first computing device that is external to an operating system of the first computing device; and

when the packet is [[not]] associated with a connection that corresponds to an application of ~~the first a second~~ computing device distinct from the first computing device, selectively encapsulating the packet and forwarding the encapsulated packet to [[a]] the second computing device in a manner transparent to the client, wherein the encapsulated packet includes a reference to a connection endpoint associated with the packet and an indication that the encapsulated packet is a forwarded packet.

20. (Previously presented) The method of claim 19 wherein the reference indicates a group of sequence numbers associated with the connection.

21. (Previously presented) The method of claim 19 wherein the reference indicates an IP address of the client, a port of an application executed by the client, an IP address of the first computing device, and a port of an application executed by the first computing device.

22. (Previously presented) The method of claim 16, further comprising:

in response to at least the initialization packet:

generating an acknowledgment to the client.

23-24. (Canceled)

25. (Currently amended) A method performed by a first computing device of an information processing system, the method comprising:

associating an application of the first computing device with a data structure associated with a connection to a client; and, ~~in a manner transparent to the client,~~

selectively:

disassociating the application of the first computing device from the data structure; and

subsequently outputting an encapsulated packet including a reference to the data structure ~~associated with the connection~~ to a second computing device to associate an application of the second computing device with the data structure, and to enable the second computing device to output packets to the client configured to appear to the client to have been sent from the first computing device.

26. (Previously presented) The method of claim 25 wherein the data structure includes a group of sequence numbers associated with the connection.

27. (Previously presented) The method of claim 25 wherein the data structure includes an IP address of the client, a port of an application executed by the client, an IP address

of the first computing device, and a port of the application executed by the first computing device.

28-30. (Canceled)

31. (Currently amended) The system of claim 1 wherein the ~~unbound data structure~~ encapsulated packet comprises a connection endpoint.

32. (Previously presented) The system of claim 1 wherein the first computing device is configured to migrate the unbound data structure by:
storing a reference to a second computing device; and
associating the stored reference with the unbound data structure.

33. (Previously presented) The system of claim 1 wherein the first computing device is configured to select the computing device to service the client based at least in part on a state of the first computing device.

34. (Previously presented) The system of claim 4 wherein the application of the first computing device is a socket-based application.

35-37. (Canceled)

38. (Currently amended) The system of claim [[37]] 10 wherein the application of the first computing device is of a first type and the application of the second computing device is of a second type.

39. (Currently amended) The system of claim [[37]] 10 wherein the first computing device is configured to selectively disassociate the application of the first computing device from the data structure based at least in part on a state of at least one of the first computing device or the second computing device.

40. (Previously presented) The method of claim 16 wherein the bound data structure comprises a connection endpoint.

41. (Previously presented) The method of claim 16 wherein the first server is configured to migrate the data structure by:

storing a reference to a second server; and

associating the reference with the unbound data structure.

42. (Previously presented) The method of claim 16 wherein the selecting the server to service the client comprises selecting a server based at least in part on a state of the first server.

43. (Previously presented) The method of claim 19 wherein the application of the first computing device is a socket-based application.

44-46. (Canceled)

47. (Currently amended) The method of claim [[46]] 25 wherein the application of the first computing device is of a first type and the application of the second computing device is of a second type.

48. (Currently amended) The method of claim [[46]] 25 wherein [[the]] selectively disassociate disassociating the application of the first computing device from the data structure comprises selectively disassociating based at least in part on a state of at least one of the first computing device or the second computing device.

49. (Currently amended) A computer-readable memory medium containing instructions for controlling a processor of a first server to selectively load balance and direct network requests among a plurality of servers by:

receiving a request packet originating from a client;

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

selectively:

when the packet is associated with a connection endpoint bound to a socket of an application of the first server, forwarding the packet and a reference to the associated connection endpoint to a protocol stack of the first server that is external to an operating system of the first server; and

when the packet is associated with a connection endpoint bound to a socket of an application of a second server, encapsulating the packet and forwarding the encapsulated packet to a second server ~~in a manner transparent to the client~~, the encapsulated packet including a reference to the associated connection endpoint bound to the socket of the application of the second server and an indication that the encapsulated packet is a forwarded packet.

50. (Previously presented) The computer-readable memory medium of claim 49 further comprising instructions that control the processor of the first server by:

including in the reference that is forwarded to the protocol stack an indication of a group of sequence numbers associated with the connection.

51. (Previously presented) The computer-readable memory medium of claim 49 further comprising instructions that control the processor of the first server by:

including in the reference of the encapsulated packet an indication of a group of sequence numbers associated with the connection.

52. (Previously presented) The computer-readable memory medium of claim 49 further comprising instructions that control the processor of the first server by:

including in the reference that is forwarded to the protocol stack an indication of an IP address of the client, a port of an application executed by the client, an IP address of the first server, and a port of an application executed by the first server.

53. (Currently amended) A computer-readable memory medium containing instructions for controlling a processor of a first server to selectively load balance and direct network requests among a plurality of servers by:

associating an application of the first server to a data structure associated with a connection with a client;

selectively disassociating the application of the first server from the data structure ~~associated with the connection~~; and

subsequently outputting an encapsulated packet including a reference to the data structure ~~associated with the connection~~ to a second server for associating an application of the second server ~~[[to]]~~ with the data structure ~~associated with the connection in a manner transparent to the client, and to enable the second server to output packets to the client configured to appear to the client to have been sent from the first server.~~

54. (Previously presented) The computer-readable memory medium of claim 53 further comprising instructions that control the processor of the first server by:

including in the data structure a group of sequence numbers associated with the connection.

55. (Previously presented) The computer-readable memory medium of claim 53 further comprising instructions that control the processor of the first server by:

including in the outputted reference an indication of a group of sequence numbers associated with the connection.

56. (Previously presented) The computer-readable memory medium of claim 53 further comprising instructions that control the processor of the first server by:

including in the outputted reference an indication of an IP address of the client, a port of an application executed by the client, an IP address of the second server, and a port of an application executed by the second server.

57. (Previously presented) The computer-readable memory medium of claim 53 further comprising instructions that control the processor of the first server by:

re-associating the application of the first server to the data structure associated with the connection with the client.

58. (Currently amended) A first server, comprising:

a memory configured to store [[a]] an unbound data structure associated with a connection to a client originating an initialization packet;

a network protocol stack external to an operating system of the first server; and

a module configured to selectively bind the data structure associated with the connection to the client to an application of the first server, and, when the first server is not selected to service the client, to migrate the data structure ~~associated with the connection in a manner transparent to the client~~ to a second server by outputting an encapsulated packet including the unbound data structure to the second server to enable the second server to output packets to the client configured to appear to the client to have been sent from the first server.

59. (Canceled)

60. (Previously presented) The information processing system of claim 1 wherein the first computing device is an intelligent network interface controller.

61. (Previously presented) The information processing system of claim 4 wherein the first computing device is an intelligent network interface controller.

62. (Previously presented) The information processing system of claim 10 wherein the first computing device is an intelligent network interface controller.

63. (Currently amended) The method of claim 16[[.]] wherein the first server comprises an intelligent network interface controller, the intelligent network interface controller performing the method on behalf of the first server.

64. (Previously presented) The method of claim 19 wherein the first computing device is an intelligent network interface controller.

65. (Previously presented) The method of claim 25 wherein the first computing device is an intelligent network interface controller.

66. (Previously presented) The computer-readable memory medium of claim 49 wherein the first server comprises an intelligent network interface controller, and the processor of the first server is a processor of the intelligent network interface controller.

67. (Previously presented) The computer-readable memory medium of claim 53 wherein the first server comprises an intelligent network interface controller, and the processor of the first server is a processor of the intelligent network interface controller.

68. (Previously presented) The first server of claim 58 further comprising an intelligent network interface controller, wherein the intelligent network interface controller comprises the memory, the network protocol stack, and the module.

69. (New) The system of claim 1 wherein the encapsulated packet includes a flag indicating that the packet is an encapsulated packet.